AUTORENTA

# 

**CAMILO MACHUCA VEGA**

**P1**

**PEDRO FELIPE GÓMEZ BONILLA**

**CAMPUSLANDS**

**ARTEMIS  
RUTA JAVA**

**FLORIDABLANCA**

**2024**

Tabla de Contenidos

[**Introducción 4**](#_xy776j9pw1un)

[**Caso de Estudio 5**](#_x5wjmxr9s7lx)

[**Planificación 5**](#_8muzr1pu3mlz)

[Construcción del Modelo Conceptual 5](#_akgvn7l12xs2)

[Descripción 5](#_qxuss0va0z8)

[Gráfica 5](#_u45t31dnlow2)

[Descripción Técnica 6](#_gww41npptmyg)

[Construcción del Modelo Lógico 6](#_nzoi4crbe63p)

[Descripción 6](#_46z7wrz43fjb)

[Gráfica 6](#_cx639fevpp57)

[Descripción Técnica 6](#_vemj55jop1dw)

[Normalización del Modelo Lógico 6](#_j975ki11pvw9)

[Primera Forma Normal (1FN) 7](#_ipsnvw5v2v0u)

[Descripción 7](#_yudex7qmv5l6)

[Gráfica 7](#_u1g2e96ymz30)

[Descripción Técnica 7](#_rjmpapoxneod)

[Segunda Forma Normal (2FN) 7](#_3fyocgo001fx)

[Descripción 8](#_n31b5bkefr23)

[Gráfica 8](#_mywvu5xf7mjo)

[Descripción Técnica 8](#_i9ctpni5mu37)

[Tercera Forma Normal (3FN) 8](#_aeip5fb7t3s5)

[Descripción 8](#_v6plqfy3484u)

[Gráfica 8](#_bcqznd6b310i)

[Descripción Técnica 8](#_qaz530fwxr49)

[Construcción del Modelo Físico 9](#_muq8ro9qbje6)

[Descripción 9](#_5g32igevz4r8)

[Código 9](#_ryjwvstap29b)

[Descripción Técnica 10](#_f9n2yx58hlja)

[Diagrama E-R 10](#_l2vl1lqjv6nt)

[Descripción 10](#_b542o9532ses)

[Gráfica 10](#_8wg77tt5rsnt)

[Descripción Técnica 10](#_whj2rpgnzs4v)

[Tablas 11](#_s3pvkylovom)

[Descripción 11](#_ye40r8z6ajh3)

[Gráfica 11](#_mgogeuideyfr)

[Descripción Técnica 11](#_4tau2idq5j2j)

[Relaciones entre Tablas 11](#_v02u5svki3o6)

[Descripción 11](#_mn4p1kzcs5d)

[Gráfica 12](#_ahrg4l6677cx)

[Descripción Técnica 12](#_v3u2pmnyvhxt)

[Inserción de Datos 12](#_khdudtfa7c4u)

[Descripción 12](#_r9p210nyg848)

[Gráfica 12](#_wpdzy6svxf2h)

[Descripción Técnica 12](#_cudxay80qe23)

[**Referencias 13**](https://docs.google.com/document/d/1SRuuADof0v8w4HWqaVgGDCH4xK5B4f5NYmcsIqbwk-s/edit?tab=t.0#heading=h.ywn691xazve)

# 

# Introducción

En el siguiente documento se presenta la realización de una base de datos la cual se relaciona con la gestión de inventario, ventas, y clientes de una tienda de libros para esto se comenzó construyendo el modelo conceptual en el cual hay cinco entidades con sus atributos y relaciones, luego realizamos el modelo lógico que consiste en la creación de tablas estas pasarían a representar las entidades del modelo conceptual y finalizamos con la creación del diagrama físico que consiste en la perfección de las tablas que realizamos en el modelo lógico.

# 

# 

# Caso de Estudio

Diseñar una base de datos para una tienda de libros que gestione el inventario, ventas y clientes. La base de datos debe permitir el registro y gestión de libros, autores, clientes, pedidos y transacciones de compra. Debes crear un diagrama UML E-R que represente la estructura de la base de datos y entregar una documentación detallada que explique las decisiones de diseño, las relaciones entre las tablas y las restricciones impuestas.

# Planificación

## Construcción del Modelo Conceptual

En este modelo, se identificaron las entidades, sus atributos y las relaciones entre ellas. Este modelo conceptual proporciona una visión detallada de cómo va a ser la base de datos además se tienen unas restricciones y relaciones.

### Descripción

Este paso tiene la estructura de la base de datos que está formada por cinco entidades y sus atributos las cuales son:

1. **Libros:**

* Id\_autor: id del autor que se relaciona con el libro.
* ISBN: Id único de libros.
* Título: nombre del libro.
* Editorial: nombre del editorial.
* Categoria: nombre de la categoría del libro.
* Fecha\_de\_publicacion: fecha en que se publicó el libro.
* Precio: Precio del libro.
* Cantidad stock: cantidad disponible del libro.

**Atributos:** Id\_ Autor, ISBN, Título, Editorial, Categoría, Fecha de publicación, precio, cantidad stock.

**Llave primaria:** ISBN

**Llave foránea**: Id\_Autor

1. **Clientes:**

* Nombre: Nombre completo del cliente.
* Teléfono: Numero del teléfono del cliente.
* Dirección: Dirección de residencia del cliente.
* Id\_cliente: id único del cliente.
* Correo\_electronico: correo personal del cliente.

**Atributos:** Nombre, Teléfono, dirección, Id\_cliente, correo electrónico

**Llave primaria:** Id\_cliente

1. **Pedidos:**

* Id\_pedidos: id único de pedidos.
* Fecha\_de\_compra: fecha en que se realizó la compra.
* Estado: estado en que se encuentra el pedido.
* Libros: libros que contiene ese pedido.
* Cantidad: cantidad de pedidos.
* Id\_cliente: id del cliente que se relaciona con el pedido.
* ISBN: ISBN de libros que se relaciona con el pedido

**Atributos:** Fecha de compras, estado, id\_pedidos, libros, cantidad, id\_cliente, ISBN.

**Llave primaria:** Id\_pedidos

**Llave foránea**: Id\_Cliente

1. **Transacciones**

* Id\_transacciones: id único de transacciones.
* Fecha\_transaccion: fecha en que se realizó la transacción.
* Total: total de la transacción.
* Método\_pago: método de pago de la transacción.
* Id\_pedidos: id del pedido que se relaciona con la transacción.

**Atributos:** Fecha de transacciones, Total, Método de pago, Id\_transacciónes, Id\_pedidos.

**Llave primaria:** Id\_transacciones

**Llave foránea**: Id\_Pedidos

1. **Autores**

* Id\_autor: id único de autores.
* Nombre: nombre del autor.
* Fecha\_nacimientro: fecha de nacimiento del autor.
* Nacionalidad: nacionalidad del autor.

**Atributos:** id\_Autor, Nombre, Fecha de nacimiento, Nacionalidad.

**Llave primaria:** Id\_Autor.

**Relaciones y Cardinalidades**

Se realizó las relaciones y cardinalidades respectivas del modelo conceptual con sus entidades

para tener mejor visualización de la base de datos:

1. **Libros- Autores:**

* Relación: “Tiene”, un libro puede tener varios autores y un autor puede tener varios libros.
* Cardinalidad: N:M (muchos a muchos).

1. **Libros- Pedidos:**

* Relación: “Incluir”, muchos libros pueden estar en un pedido y un pedido puede tener muchos libros.
* Cardinalidad: N:1 (muchos a uno).

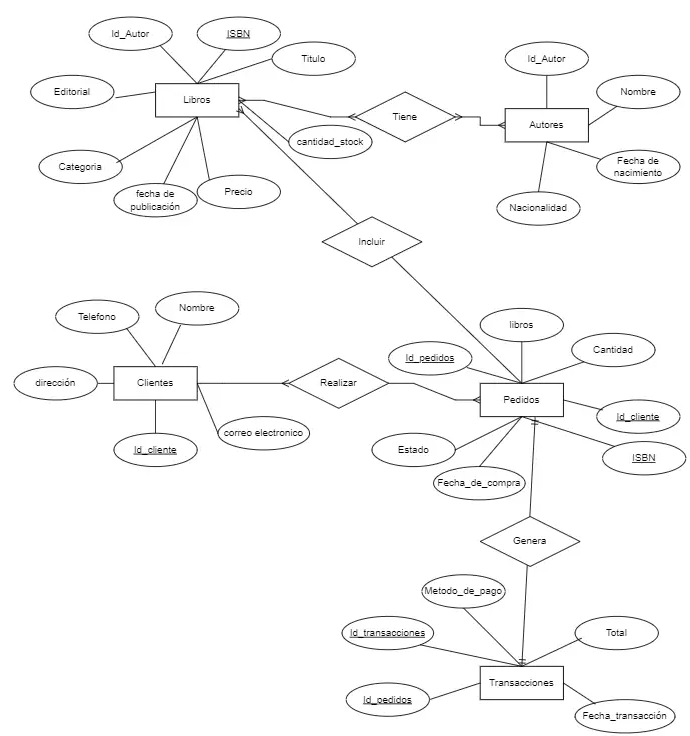
1. **Pedidos- Clientes:**

* Relación: “Realizar”, muchos pedidos pueden ser de un cliente y un cliente puede realizar muchos pedidos.
* Cardinalidad: N:1 (muchos a uno).

1. **Pedidos- Transacciones:**

* Relación: “Generar”, un pedido genera una transacción y una transacción se genera con un pedido.
* Cardinalidad: 1:1 (uno a uno).

### Gráfica



### Descripción Técnica

Estas entidades tienen unas restricciones y relaciones las cuales están entre libros y autores, entre clientes y pedidos, y entre pedidos y transacciones. Entre libros y autores hay una relación de muchos a muchos ya que un libro puede tener muchos autores y un autor puede tener muchos libros, entre libros y pedidos hay una relación de muchos a uno ya que un pedido puede tener muchos libros y un libro está en un pedido, entre clientes y pedidos hay una relación de muchos a uno ya que un cliente puede realizar muchos pedidos.

## Construcción del Modelo Lógico

En este modelo se realiza la conversión de las entidades a tablas las cuales no pueden tener una relación de muchos a muchos por ende se realizaron una subtablas u añadieron nuevas entidades.

### Descripción

Este paso tenemos las tablas de las entidades de la base de datos que está formada por siete entidades y sus atributos las cuales son:

1. **Libros:**

* Libros(ISBN INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Titulo VARCHAR(25) NOT NULL,
* Cantidad\_stock INT NOT NULL,
* Precio INT NOT NULL,
* FechaPublicación DATE NOT NULL,
* Categoria VARCHAR(30) NOT NULL,
* Editorial VARCHAR(30) NOT NULL

**Atributos:** Id\_Solicitud, ISBN, Título, Editorial, Categoría, Fecha de publicación, precio, cantidad stock.

**Llave primaria:** ISBN

**Llave foránea**: Id\_solicitud

1. **Clientes:**

* Id\_Cliente INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Nombre VARCHAR(30) NOT NULL,
* Teléfono VARCHAR(15),
* CorreoElectrónico VARCHAR(35),
* Id\_direccion INT,
* FOREIGN KEY (Id\_direccion) REFERENCES Dirección(Id\_direccion)

**Atributos:** Nombre, Teléfono, Id\_dirección, Id\_cliente, correo electrónico

**Llave primaria:** Id\_cliente

**Llave foránea**: Id­\_dirección

1. **Pedidos:**

* Id\_pedidos INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Estado VARCHAR(25) NOT NULL,
* Fecha\_de\_compra DATE NOT NULL,
* Cantidad INT NOT NULL,
* Id\_cliente INT NOT NULL,
* ISBN INT NOT NULL,
* FOREIGN KEY (Id\_Cliente) REFERENCES Cliente(Id\_Cliente),
* FOREIGN KEY (ISBN) REFERENCES Libros(ISBN)

**Atributos:** Fecha de compra, estado, id\_pedidos, libros, cantidad, id\_cliente, ISBN.

**Llave primaria:** Id\_pedidos

**Llave foránea**: Id\_Cliente, ISBN

1. **Solicitud**

* Id\_solicitud INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Id\_Autor INT NOT NULL,
* ISBN INT NOT NULL,
* FOREIGN KEY (Id\_Autor) REFERENCES Autores(Id\_Autor),
* FOREIGN KEY (ISBN) REFERENCES Libros(ISBN)

**Atributos:** Id\_solicitud, Id\_autor, ISBN.

**Llave Primaria:** Id\_Solicitud

**Llave foránea:** Id\_Autor, ISBN

1. **Dirección**

* Dirección(Id\_direccion INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Pais VARCHAR(25) NOT NULL,
* Ciudad VARCHAR(25) NOT NULL,
* Municipio VARCHAR(25) NOT NULL

**Atributos:** Id\_dirección, país, ciudad, Municipio.

**Llave primaria:** Id\_dirección

1. **Transacciones**

* Id\_Transacciones INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Método\_de\_pago VARCHAR(25) NOT NULL,
* Total INT,
* Fecha\_transacción DATE NOT NULL,
* Id\_pedidos INT,
* FOREIGN KEY (Id\_pedidos) REFERENCES Pedidos(Id\_pedidos)

**Atributos:** Fecha de transacciones, Total, Método de pago, Id\_transacciónes, Id\_pedidos.

**Llave primaria:** Id\_transacciones

**Llave foránea**: Id\_Pedidos

1. **Autores**

* Id\_autor: INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,
* Nombre: VARCHAR(25) NOT NULL,
* FechaNacimiento: DATE NOT NULL,
* Nacionalidad: VARCHAR(25) NOT NULL

**Atributos:** id\_Autor, Nombre, Fecha de nacimiento, Nacionalidad.

**Llave primaria:** Id\_Autor

**Relaciones y Cardinalidades**

Se realizó las relaciones y cardinalidades respectivas del modelo lógico con sus entidades para tener mejor visualización de la base de datos:

1. **Autores- solicitud:**

* Un autor puede tener muchas solicitudes y muchas solicitudes pueden tener un autor. 1: N (uno a muchos).



1. **Solicitud- Libros:**

* Muchas solicitudes puden tener un libro y un libro puede estar en muchas solicitudes. 1: N (uno a muchos).



1. **Libros- Pedidos:**

* Un pedido puede tener muchos libros y muchos libros pueden estar en un pedido 1: N (uno a muchos).



1. **Pedidos- clientes:**

* Muchos pedidos pueden estar con un cliente y un cliente puede hacer muchos pedidos 1: N (uno a muchos).



1. **Clientes- Direccion:**

* Un cliente puede tener una dirección y una dirección puede tener un cliente. 1: 1 (uno a uno).

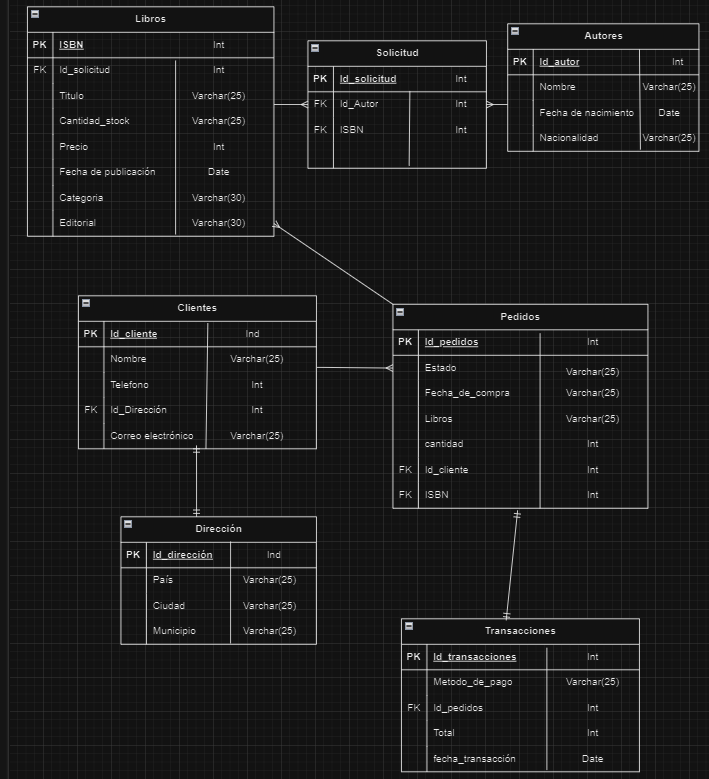


1. **Pedidos- Transacciones:**

* Un pedido puede tener una transacción y una transacción puede tener un pedido. 1:1 (uno a uno).



### Gráfica



### 

### Descripción Técnica

Estas entidades tienen unas relaciones las cuales están entre libros y autores, entre clientes y pedidos, y entre pedidos y transacciones. Entre libros y autores hay una relación de muchos a muchos ya que un libro puede tener muchos autores y un autor puede tener muchos libros, entre libros y pedidos hay una relación de muchos a uno ya que un pedido puede tener muchos libros y un libro está en un pedido, entre clientes y pedidos hay una relación

## Normalización del Modelo Lógico

Se realizó el proceso de la normalización de las tablas anteriormente visualizadas para organizar sus datos de la manera más eficiente. En esta parte se organizaron los datos para reducir su redundancia y dependencias transitivas en la base de datos.

### Primera Forma Normal (1FN)

En esta forma se necesita que los atributos se encuentren atómicos en la mínima expresión que no haya más de dos datos en cada columna como se observa en la gráfica 2

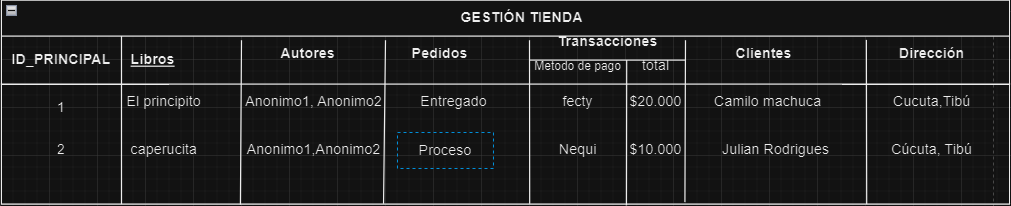
Una tabla esta en 1FN si cumple con los siguiente reglas o criterios:

* Todos los atributos contienen valores atómicos (indivisibles).
* No debe haber grupos repetidos de columnas.
* Cada columna debe contener un solo valor en cada fila.

#### Descripción

Se realizó una tabla de la gestión de una tienda en este ejemplo podemos observar que hay unos datos que no cumplen con las reglas (1FN) los cuales son los autores ya que hay dos actores en la misma columna lo que no debería ser posible ya que los datos deben ser atómicos y deben cumplir con las reglas de 1FN como la gráfica 2.

#### Gráfica





#### Descripción Técnica

1. **Autores:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

1. **Solicitud:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

1. **Libros:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

1. **Pedidos:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

1. **Clientes:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

1. **Transacciones:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

1. **Dirección:**

* Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.

### Segunda Forma Normal (2FN)

Una tabla está en 2FN si cumple con las siguientes reglas o criterios:

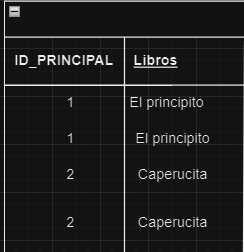
* Está en 1FN.
* Todos los atributos no claves (no pertenecientes a una clave primaria compuesta dependen de la clave primaria u de la clave de la tabla principal.)

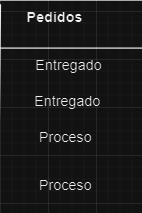
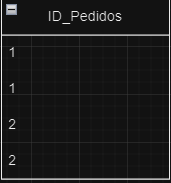
#### Descripción

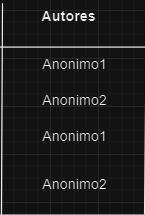
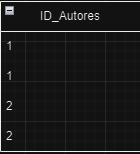
La segunda forma normal, es el segundo nivel de normalización en el diseño de la base de datos que se aplicara a las tablas de la base de datos que ya cumplen con la primera forma normal y lleva a cabo la eliminación de dependencias parciales dentro de una table.

En este caso de estudio los datos se separan en tablas distintas para tener una mayor organización con esto se puede desplazar a las tablas y buscar el atributo que se requiera, se dividieron en las siguientes tablas:

#### Gráfica







#### Descripción Técnica

1. **Autores:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

1. **Solicitud:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

1. **Libros:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

1. **Pedidos:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

1. **Clientes:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

1. **Transacciones:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

1. **Dirección:**

* Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

### Tercera Forma Normal (3FN)

Una tabla está en 3NF si cumple con los siguientes criterios:

* Está en 2NF.
* No hay dependencias transitivas: ningún atributo no clave depende de otro atributo no clave.

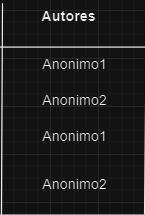
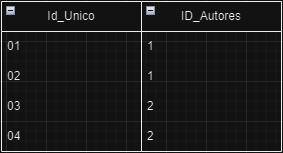
En esta forma se verifica que cada tabla creada en la forma Normal 2FN sea independiente de la tabla principal ya que no deben existir atributos no principales que dependan transitivamente del atributo clave.

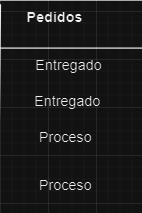
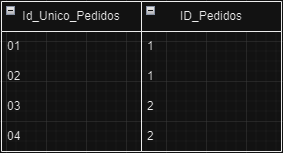
#### Descripción

La tercera forma normal, es el tercer nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de una base de datos que ya cumplen con la segunda forma normal

En este caso las tablas tienen un id único o llave primaria para que no tenga una dependencia con la tabla principal como se muestra en las siguientes tablas:

#### Gráfica





#### 

#### Descripción Técnica

1. **Autores:**

* Se encuentra en 3FN, ya que esta en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

1. **Solicitud:**

* Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

1. **Libros:**

* Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

1. **Pedidos:**

* Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

1. **Clientes:**

* Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

1. **Transacciones:**

* Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

1. **Dirección:**

* Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

## Construcción del Modelo Físico

Este modelo es la última etapa de la creación de diseño de la base de datos en el cual se tiene en consideración el modelo lógico, en la cual se verificaron las restricciones y se especificaron las estructuras de las tablas. Además, este modelo incorpora los tipos de datos de los atributos previamente definidos, los cuales fueron estructurados en tablas utilizando el lenguaje de un sistema de gestión de bases de datos (SGBD) compatible con la plataforma MySQL.

### Descripción

El modelo físico se diseñó para funcionar en MySQL, donde cada entidad se representa como una tabla compuesta por sus atributos correspondientes como también que tipo de dato contiene cada atributo ya sea INT, VARCHAR, DATE, entre otros.

### Código

Para crear la base de datos utilice el siguiente comando:

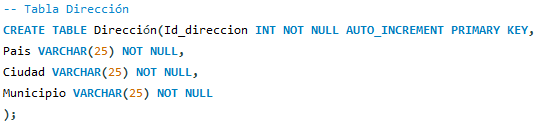


Para utilizar la base de datos ocupe el siguiente comando:

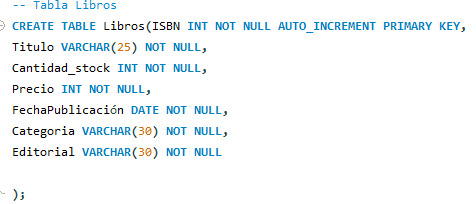


Comenzamos creando las tablas junto con sus tipos de datos correspondientes. Para esto, se utilizan los siguientes comandos:

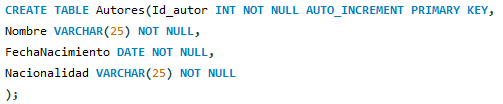
1. Creación de la tabla Dirección



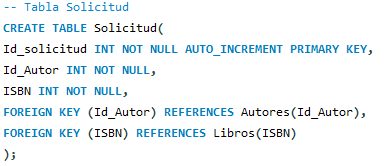
1. Creación de la tabla Libros



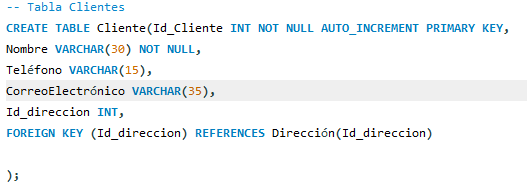
1. Creación de la tabla Autores



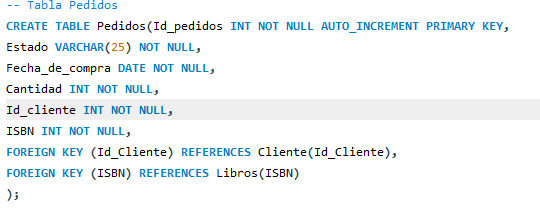
1. Creación de la tabla Solicitud



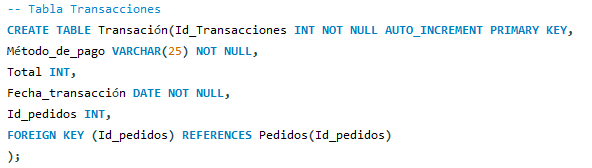
1. Creación de la tabla Clientes



1. Creación de la tabla Pedidos



1. Creación de la tabla Transacciones



Finalmente, para listar las tablas anteriormente creadas utilizamos este comando:



**Construcción del Diagrama UML**

Se ha diseñado un diagrama UML tomando como referencia la normalización para entender mejor los diseños, la arquitectura del código y la implementación propuesta. Este enfoque nos permitirá tener una visión clara y detallada de cómo se manejarán cada una de las consultas, funcionalidades y los usuarios en la base de datos. De esta manera, podremos asegurar una implementación eficiente y coherente con los requisitos del sistema.

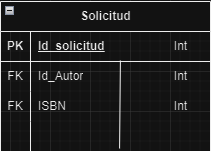
**Descripción**

El diagrama UML se ha diseñado con el objetivo de representar detalladamente la estructura de cada tabla y sus relaciones. Este diagrama da a conocer claramente el tipo de dato correspondiente a cada atributo, así como la identificación de claves primarias (primary Keys) y claves foráneas (foreign Keys). Además, se especifica la obligatoriedad de los atributos, proporcionando una visión precisa y exhaustiva de la base de datos. Esto facilita el entendimiento y la implementación técnica, con esto se asegura que todos los componentes y sus interrelaciones estén correctamente definidos.

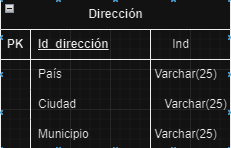
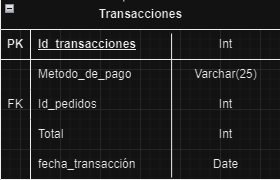
Comenzaremos creando las tablas junto con sus tipos de datos correspondientes:

1. Tabla Autores
2. Tabla Solicitud
3. Tabla Libros
4. Tabla Pedidos
5. Tabla Clientes
6. Tabla Transacciones
7. Tabla Direcciones

###### Gráfica







Inserciones de Datos

La inserción de datos en las tablas es una parte importante para poder realizar la prueba de operatividad.

Para insertar datos en una tabla en especifica se utiliza la siguiente sintaxis:



###### Descripción Técnica

##### Relaciones entre Tablas

Se da a visualizar las relaciones que hay entre las entidades o tablas por medio de

###### Descripción

En esta parte se subrayaron las relaciones entre tablas de diferentes colores cada relación.

###### Gráfica

